Senior Design
ENG EC 463

Test Report - Prototype 1.0

Team 21 - IoT Kitchen

Team Members
Addison Dolido addison@bu.edu
Erin Dorsey edorsey1@bu.edu
Saransh Kothari saranshk@bu.edu
Yuran Shi yuran@bu.edu
Kenny Zheng ykzheng2@bu.edu

Submitted: November 17th, 2019

Scale Weight Sensitivity

Equipment and Setup

The equipment required for this test are the "smart" scale (consisting of an arduino Uno, an HC-05 bluetooth module, a 5-kg load cell, an HX711 amplifier, some wood, and an LCD screen), a set of weights, a USB cable Type A/B, and a computer with an appropriate USB port. To begin, one must place the scale on a flat surface and plug it into the computer. Then the LCD screen turns on and the user must wait until it reads "Ready for weight". At this point the test can be started.

The test consisted of placing a known weight and recording what the scale measured. This was repeated several times. Then the weight was incrementally removed until the lowest weight was removed and the weight present on the scale when it read 0 was recorded.

Results

The measurable criteria were as follows:

1. Percent difference between known weight and scale measurement
2. The smallest weight the scale registers

Known weight (g)	Measured weight (g)	Percent difference (%)
500	500	0
520	520	0
521	521	0
571	571	0
71	71	0
21	21	0
1	1	0
0	0	0

Conclusions

This test was very much a success. Our scale read the correct weight to the nearest gram for all of our masses used. Plus, the lowest registered weight was 1 gram and the scale only returned back to 0 once all of the weight was removed. This test verifies the accuracy and sensitivity of our initial scale design. Thus, our scale is accurate enough to measure to the nearest gram, which is accurate enough to measure most food ingredients including lettuce, which weighs over 30 grams per cup.

Scale Taring

Equipment and Setup

The equipment required for this test are the “smart” scale (consisting of an arduino Uno, an HC-05 bluetooth module, a 5-kg load cell, an HX711 amplifier, some wood, and an LCD screen), a set of weights, a USB cable Type A/B, and a computer with an appropriate USB port. To begin, when has to place the scale on a flat surface and plug it into the computer. Then the LCD screen turns on and the user must wait until it reads “Ready for weight”. Finally, the arduino serial monitor must be opened on the computer. At this point the test can be started.

The test consisted of placing some weight on the scale and then in the serial monitor on the computer pressing “t” and hitting enter. This then tared the scale, which should now read a weight of 0. Finally, a known weight was added to the scale and the measured weight was recorded. This entire procedure was repeated several times.

Results

The measurable criteria were as follows:

1. Is user able to get scale back to 0 after taring **(Yes/No)**
2. Percent difference between expected measurements after taring and actual scale reading

Every time after the scale was tared, the measurement of the scale was 0.

Measurements after taring:

Known weight (g)	Measured weight (g)	Percent difference (%)
50	50	0
20	20	0
500	500	0
1	1	0
2	2	0
500	500	0

520	520	0
20	20	0

Conclusions

This test was another success. The scale was able to consistently tare properly, with multiple different weights. Plus, after each tare the scale continued to read weights accurately to the nearest gram. This proves that our idea of the user taring with the bowl on the scale and then adding an ingredient is plausible with our current plan.

Bluetooth Signaling

Equipment and Setup

The equipment required for this test is an Android device with Bluetooth connectivity, “smart” scale (consisting of an arduino Uno, an HC-05 bluetooth module, a 5-kg load cell, an HX711 amplifier, some wood, and an LCD screen), a set of weights, a USB cable Type A/B, a computer with an appropriate USB port, and our Android Studio SDK. The test was conducted by the scale being placed on a flat surface, plugged in to the computer with the Arduino Serial Monitor opened. Then open the Android mobile application which will request an option for connection to Bluetooth. By doing so, the device will connect to the HC-05 module on the Arduino Scale.

Results

Once the device and scale are connected and paired via Bluetooth, “connected” should display in a textbox. You can add weight to the scale and then in the serial monitor on the computer type “q” and hit enter which will then display the weight of the measurements on the Android device.

The measurable criteria were as follows:

1. User able to connect to Arduino via Bluetooth **(Yes/No)**
2. User able to see Arduino Scale Data **(Yes/No)**

Conclusions

The bluetooth signaling was able to succeed all test criterias. The application was able to connect to Arduino Scale via bluetooth and display the data without crashing. Thus, our choice in using bluetooth sensors is confirmed to work for our needs.

Authentication

Equipment and Setup

The equipment required for this test is an Android Device. The software was implemented using Android Studio and Google Firebase. The test was conducted by connecting the mobile device with our application to the internet and using google sign in option to login to the account. Once user is logged in, the application will take the user to the bluetooth page with the logout button, which user can log out their account.

Results

Google firebase console will be used to check whether users login and logout successfully.

The measurable criteria were as follows:

1. User able to login and log out to their google account with firebase**(Yes/No)**
2. After successful login, the application take user to the bluetooth page**(Yes/No)**
3. After successful log out, the application take user back to log in page**(Yes/No)**

Conclusions

Dialogflow Speech Recognition

Equipment and Setup

The equipment required for this test are a computer with internet connectivity, Google Dialogflow console and the Google Assistant emulator. The test was conducted by loading the Dialogflow console at https://dialogflow.cloud.google.com and selecting the iotk-nlu-test agent. Then the Google Assistant emulator is loaded by selecting "Google Assistant". The Dialogflow powered agent was launched by pressing the microphone button and verbally inputting "Talk to my test app". Test phrases were also entered by pressing the microphone button and providing verbal input.

Results

The measurable criteria were as follows:

1. Module has response for all phrases **(Yes/No)**
2. Response matches user intent **(Yes/No)**
3. Module does not crash **(Yes/No)**

For test 1, the test would be considered a success if all phrases were matched to a response that was not the default fallback response ("I didn't understand." or "Could you repeat that please?") given if the agent did not understand or have a response to input . Test 2 would be considered a success if each test phrase matched to the correct intent listed in the test plan. Finally, test 3 would be considered a success if the module did not crash at any point, even when faced with test phrases intentionally designed to push the limits of the program. These test phrases included asking the agent for ingredients and instructions that were not included in the test recipe.

Conclusions

The Dialogflow agent was able to succeed all tests and achieve the test criteria. The module was able to provide the correct response to all phrases and did not crash at any point. The module was also able to achieve a few impromptu tests, such as responding to different voices giving test phrases, and matching test phrases in a variety of volume levels. This successful test confirms the viability and usability of Dialogflow for the project, so the next steps will be integrating the module with the main app.